A Comparative Analysis of Transformer and Latent Factor Models with Sentimental Analysis for Rating Prediction in Recommender Systems

Jinlong Ruan
j1ruan@ucsd.edu
University of California, San Diego
San Diego, California, USA

Zeyu Wang
zew027@ucsd.edu
University of California, San Diego
San Diego, California, USA

Xintong Chen
xic060@ucsd.edu
University of California, San Diego
San Diego, California, USA

Ran Ji
raji@ucsd.edu
University of California, San Diego
San Diego, California, USA

ABSTRACT

This work conducts a comprehensive comparative analysis of Transformer and Latent Factor Models, integrated with sentiment analysis, for rating prediction in recommender systems. We employ a rich dataset of Google Restaurant reviews, which includes textual feedback and numerical ratings, to explore the correlation between sentiment scores and customer ratings. Our study reveals that while Transformers excel at capturing nuanced relationships within text sequences, they demand considerable computational resources and are not well suited to the rating prediction task. On the other hand, Latent Factor Models, especially when extended to include sentiment scores from review texts, demonstrate superior computational efficiency and robust prediction accuracy. By incorporating sentiment analysis, the Latent Factor Model offers a more nuanced understanding of user feedback, enhancing its predictive performance. The findings suggest that Latent Factor Models augmented with sentiment analysis present a more effective approach for rating prediction in the context of recommender systems, balancing computational efficiency with predictive accuracy and a deeper understanding of user sentiment.

KEYWORDS

Recommender Systems, Rating Prediction, Transformer, Latent Factor Model, Sentiment Analysis

1 INTRODUCTION & DATASET

The dataset we selected for our study is a publicly accessible collection of Google Restaurant textual reviews [13], encompassing user feedback on services or products. Each review includes a text entry and a corresponding numerical rating. The dataset is divided into training, validation, and testing sets, comprising 87013, 10860, and 11015 reviews, respectively. These reviews provide not only immediate user feedback but also carry unique identifiers for both users and merchants.

A preliminary exploratory analysis revealed the textual richness of the dataset, with 97459 unique words, and a distribution of ratings that offers foundational insights for model design. Firstly, sentiment analysis, as shown in Figure 1, conducted with NLTK's Sentiment Intensity Analyzer on customer ratings shows a close correlation between higher ratings and more positive sentiment scores. This trend is more pronounced at higher ratings and more positive sentiment scores. This trend is more pronounced at higher rating levels, suggesting a prevalence of positive feedback in higher ratings. In contrast, lower ratings display a variety of sentiments, while higher ratings demonstrate consistent positive sentiment with fewer outliers.

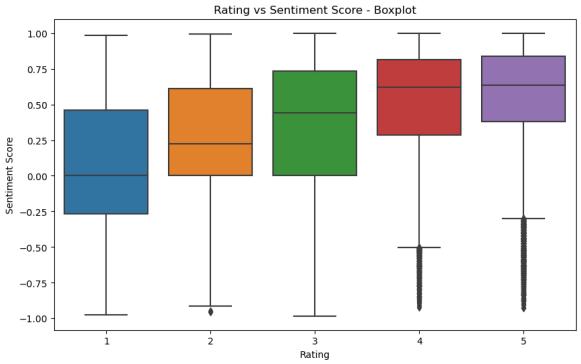

Figure 1: Rating vs Sentiment.

Meanwhile, the generated overall words cloud in Figure 2 vividly depicts the most frequently mentioned words in the reviews, particularly the prominent positioning of positive emotional words such as "great," "good," "love," "amazing," "best," and "delicious," indicating a generally positive trend in customer feedback. Food-related words such as "pizza," "chicken," "steak," "salad," "shrimp," and "taco" are also prominent in the word cloud, possibly reflecting their common mention in customer reviews and the importance to customer experience. Descriptive words related to taste and texture, such as "tasty," "fresh," "crispy," and "tender," further emphasize customers' attention to food quality. Overall, this word cloud provides us with an intuitive understanding of customer preferences and restaurant performance.

This analysis offers key perspectives for businesses to assess customer satisfaction and service quality, playing a significant role in informing business strategy and enhancing customer relations.

Figure 2: Overall Words Cloud.

2 PREDICTIVE TASK

The objective is to predict the final rating based on information from a review, which includes a user, a business, the review text, pictures, and the user's history reviews. Among these features, we only picked user-business interactions and review texts as our available options since pictures provide limited information about ratings, and history reviews can be reflected in the entire dataset. Specifically, in our two methods, the text-based transformer only uses the text as its input, on the other hand, the latent factor model exploits both interactions and texts. Due to the simplicity of our required features, we did not need to apply any pre-processing to the data. Details about how data is transformed in our methods are elaborated in the methodology section.

We measured the mean squared error (MSE) as Equation 1 to determine the validity of our predictions. MSE is suitable for this task since all ratings are in a range of 1 to 5, therefore outliers will not significantly affect the error.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(p_i - r_i)^2 \tag{1}$$

We observed that the dataset is skewed, in which most ratings are in a range of 3 to 5. A trivial predictor that always predicts the rating as 4 can lead to a decent MSE. We selected this model as our baseline and compared our methods against it.

3 METHODOLOGY

3.1 Transformer

The Transformer is one of the most popular models employed for natural language processing in recent years. The architecture of the Transformer model is depicted in Figure 3 as in the original paper of Vaswani et al. [11].

The Transformer model can capture intrigued relationships in the input text sequences by its self-attention mechanism. For each input vector X_i, we have three vectors: the query vector Q_i, key vector K_i and the value vector V_i.

$$\begin{aligned} Q &= XW^Q \\ K &= XW^K \\ V &= XW^V, \end{aligned} \tag{2}$$

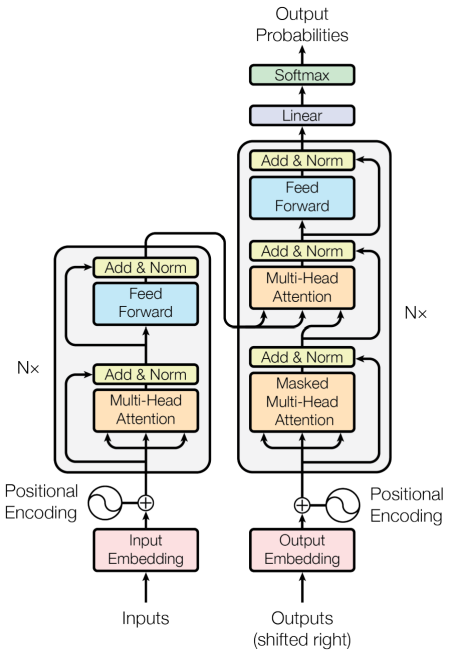

Figure 3: The Transformer - model architecture.

where $W^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W^K \in \mathbb{R}^{d_{model} \times d_k}$, and $W^V \in \mathbb{R}^{d_{model} \times d_v}$ are weight matrices to be learned in the training process of the model. The attention of each token X_i is computed as the weighted sum of all values V_j, whose weight is given by the dot product of Q_i and K_j normalized by softmax:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{3}$$

This mechanism allows the model to compute the significance of different parts of the input sequence simultaneously. The positional encoding mechanism employed by the Transformer model allows each position in the input to relate to other positions. To compute the positional information within the input sequences, the model uses a vector with the same dimension as the embedding vectors, d_{model}. The positional encoding is given by:

$$\begin{aligned} PE_{(pos,2i)} &= \sin(pos/10000^{2i/d_{model}}) \\ PE_{(pos,2i+1)} &= \cos(pos/10000^{2i/d_{model}}), \end{aligned} \tag{4}$$

where i is the index of the embedding vector and pos is the token's absolute position in the sequence. Combining the self-attention and the positional encoding mechanisms makes the Transformer model great at capturing long-range dependencies and contextual information in large datasets. Thus, the Transformer model is particularly effective for tasks that require an understanding of the global context [11].

3.2 Latent Factor Model

A standard form of latent factor model can be expressed as Equation 5.

$$p(u, i) = \alpha + \beta_u + \beta_i + \gamma_u \cdot \gamma_i \tag{5}$$

α is a global bias, β_u, β_i are user-specific and item-specific bias, and γ_u, γ_i are user-specific and item-specific latent embedding vectors respectively.

We also extended the model to incorporate the sentiment of the review text. It can be defined as Equation 6,

$$p(u, i, t) = \alpha + \beta_u + \beta_i + \gamma_u \cdot \gamma_i + wS(t) \tag{6}$$

where w is a float-point weight for the sentiment score, S is a sentiment analyzer that predicts the sentiment as a value between -1 and 1, and t is the review text. We utilized VADER [4] to extract sentiment scores from texts.

We adopted stochastic gradient descent to optimize the parameters. The batch size is set to one for easy implementation. In addition, we also applied momentum to avoid local minima and regularization to prevent overfitting. The update process can be described as Equation 7,

$$\begin{aligned}
err &= p(u, i, t) - r \\
m_\alpha &= -lr \cdot err + \omega \cdot m_\alpha \\
m_w &= -lr \cdot (err \cdot S(t)) + \omega \cdot m_w \\
m_{\beta_u} &= -lr \cdot (err + \lambda \beta_u) + \omega \cdot m_{\beta_u} \\
m_{\beta_i} &= -lr \cdot (err + \lambda \beta_i) + \omega \cdot m_{\beta_i} \\
m_{\gamma_u} &= -lr \cdot (err \cdot \gamma_i + \lambda \gamma_u) + \omega \cdot m_{\gamma_u} \\
m_{\gamma_i} &= -lr \cdot (err \cdot \gamma_u + \lambda \gamma_i) + \omega \cdot m_{\gamma_i} \\
\alpha &= \alpha + m_\alpha \\
w &= w + m_w \\
\beta_u &= \beta_u + m_{\beta_u} \\
\beta_i &= \beta_i + m_{\beta_i} \\
\gamma_u &= \gamma_u + m_{\gamma_u} \\
\gamma_i &= \gamma_i + m_{\gamma_i}
\end{aligned} \tag{7}$$

where m is the momentum of each parameter, ω is the weight of momentum, and λ is the weight of regularization.

3.3 VADER Sentiment Analysis

VADER is a parsimonious rule-based sentiment extraction model for informal text, more specifically, social media text that includes review text [4]. Vader can compute a sentiment score for each input sequence. The sentiment score is a real value between -1 to 1, where a sentiment score smaller than -0.05 represents negative sentiment, a sentiment score between -0.05 to 0.05 represents neutral sentiment, and a sentiment score greater than 0.05 represents positive sentiment. Due to the nature of rating and review sentiment and the fact that VADER sentiment is sensitive to both the polarity and the intensity of sentiments expressed in the input text sequences, we will be inclined to assign a higher predicted rating for a higher sentiment score and a lower predicted rating for a lower sentiment score for a given review text.

4 RELATED WORKS

4.1 Dataset Source and Usage

The dataset we used originates from public resources containing a large number of user text reviews and ratings on catering services. Similar datasets, such as the Amazon Review Dataset [12] and Yelp Review Dataset [15], have also been widely used for studying user evaluations and recommendation systems. These datasets are typically utilized for analyzing consumer behavior, sentiment analysis, and text mining to enhance service quality and personalized recommendations.

4.2 Traditional Rating Prediction Methods

Rating prediction is a key task in recommendation systems. Koren et al. (2009) proposed a matrix decomposition-based method that has been widely used for rating prediction [7]. Additionally, the Factorization Machine (FM) model by Rendle (2010) is another popular method for rating prediction, merging multiple features to improve accuracy [8]. Bell and Koren (2007) demonstrated the versatility of rating prediction by integrating various models in the Netflix competition [1].

4.3 Advantages of Latent Factor Models in Rating Prediction

Latent Factor Models (LFM), as demonstrated by Koren (2008) [6] and Salakhutdinov & Mnih (2008) [9], predict ratings by uncovering hidden relationships between users and items. He et al. (2017) further enhanced the performance of latent factor models in recommendation systems by combining deep learning and collaborative filtering techniques [3]. Particularly in our task, latent factor models excel not only in extracting deep relationships between users and items but also in higher computational efficiency when dealing with complex user review data. Harald Steck (2013) emphasized the limitations of traditional Root Mean Square Error (RMSE) evaluation metrics when facing data sparsity and selection bias [10]. In contrast, our use of Latent Factor Model (LFM) seeks to reveal the latent relationships between review texts and ratings, capturing the complex interactions between user preferences and item attributes.

4.4 Limitations of Text-Based Rating Prediction with Transformers

The Transformer model (Vaswani et al., 2017) [11] and BERT (Devlin et al., 2019) [2] have achieved significant success in the field of text processing. However, despite their success in natural language processing, these models show certain limitations when dealing with specific types of review data, such as catering service reviews. Yang et al. (2019), while proving the effectiveness of Transformers in text classification, highlighted the potential inefficiencies of these text-based methods in terms of computational resources and time costs when processing large-scale datasets [14].

5 EXPERIMENT RESULTS

5.1 Transformer

The transformer model is implemented following the original implementation with some architectural modifications for adaptation to the rating prediction task [11]. The input sequences are the review texts and target labels are the ratings corresponding to the texts. The input texts are first tokenized to build the vocabulary of size 97,459. Then, each input sequence is padded into a feature vector of size 256, and the dataset is divided into a training set of size 87,013, a validation set of size 10,860, and a testing set of size 11,015. The

target labels are ratings from 1 to 5. To have the model output a real number prediction, we set the output feature size of the last MLP layer to be 1. The d_{model} is set to 250 and the number of heads is set to 5. The dropout ratio is 0.5 and the number of layers is 1. We employ the Mean Squared Error loss function and Adam as our optimizer with a learning rate set to 0.001 [5]. We train the model for 5 epochs. The training and validation loss is depicted in Figure 4. The performance on the test set is in Table 1.

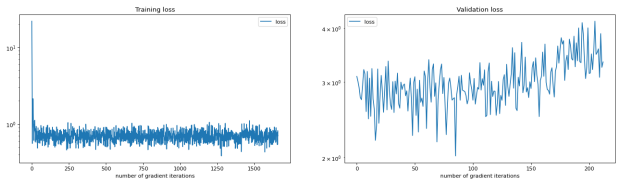

Figure 4: The Transformer - Training and Validation Loss.

5.2 Latent Factor Model

We implemented both the vanilla latent factor model and our extended model. The inputs to the vanilla latent factor model are a user's id and a business's id. In the extended model, an additional input is a sentiment score, which can be obtained from the compound score of VADER [4]. We also applied momentum to better optimize the parameters. The training and validation MSE is shown in Figure 5. The models with the lowest MSE on the validation set are selected and their performance on the test set is presented in Table 1.

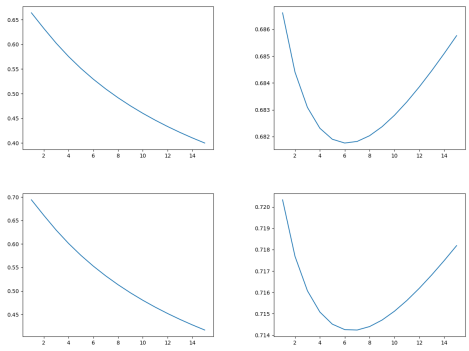

Figure 5: Training and validation MSE of vanilla and our extended latent factor model

Table 1: Experiment Result

Model	MSE
Baseline	0.904
Transformer	3.399
Vanilla Latent Factor Model	0.678
Extended Latent Factor Model	0.648

6 CONCLUSION & DISCUSSION

This study embarked on an evaluative journey to compare the efficacy of Latent Factor Models and text-based methods, particularly Transformers, in the context of rating prediction within recommender systems. Our findings illuminate the strengths and shortcomings inherent in each approach, highlighting the nuanced dynamics of predictive modeling.

Foremost, the Latent Factor Model demonstrated a robust capacity for accurate rating predictions. Its efficiency is particularly noteworthy, as it capably integrates relevant user-business interactions and review texts. This model's proficiency stems from its adept handling of these complex datasets, with minimal computational demand compared to more text-intensive approaches. By incorporating sentimental analysis, the Latent Factor Model gains an enhanced dimension, enabling it to capture subtle nuances within user feedback. This integration not only elevates the predictive accuracy but also enriches the model's understanding of user preferences and sentiments, a critical aspect in the domain of personalized recommendations.

Conversely, the Transformer-based text method, despite its advanced capabilities in natural language processing, appears less suited for this specific task. The primary drawback lies in its considerable computational resource demands, a critical factor when processing extensive datasets typical in recommender systems. Moreover, the model's inherent design, though groundbreaking for textual analysis, does not align seamlessly with the unique requirements of rating prediction, where succinct and direct user feedback is paramount.

In summary, our exploration reveals the Latent Factor Model, especially when augmented with sentimental analysis, as a superior choice for rating prediction tasks in recommender systems. Its balance of computational efficiency and predictive accuracy, coupled with the nuanced understanding of user sentiments, presents a compelling case for its application. In contrast, the text-based Transformer model, while formidable in its domain, may not be the most resource-efficient or directly applicable method for this specific predictive task.

REFERENCES

[1] Robert M. Bell and Yehuda Koren. 2007. Lessons from the Netflix Prize Challenge. *SIGKDD Explor. Newsl.* 9, 2 (dec 2007), 75–79. https://doi.org/10.1145/1345448.1345465

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[3] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web* (Perth, Australia) *(WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 173–182. https://doi.org/10.1145/3038912.3052569

[4] C. Hutto and Eric Gilbert. 2014. VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. *Proceedings of the International AAAI Conference on Web and Social Media* 8, 1 (May 2014), 216–225. https://doi.org/10.1609/icwsm.v8i1.14550

[5] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6980

[6] Yehuda Koren. 2008. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Las Vegas, Nevada, USA) *(KDD '08)*. Association for Computing Machinery, New York, NY, USA, 426–434. https://doi.org/10.1145/1401890.1401944

[7] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37. https://doi.org/10.1109/MC.2009.263

[8] Steffen Rendle. 2010. Factorization Machines. In *2010 IEEE International Conference on Data Mining*. 995–1000. https://doi.org/10.1109/ICDM.2010.127

[9] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. 2007. Restricted Boltzmann machines for collaborative filtering. *ACM International Conference Proceeding Series* 227, 791–798. https://doi.org/10.1145/1273496.1273596

[10] Harald Steck. 2013. Evaluation of recommendations: rating-prediction and ranking. In *Proceedings of the 7th ACM conference on Recommender systems*. 213–220.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[12] Mengting Wan and Julian McAuley. 2016. Modeling ambiguity, subjectivity, and diverging viewpoints in opinion question answering systems. In *International Conference on Data Mining (ICDM)*.

[13] An Yan, Zhankui He, Jiacheng Li, Tianyang Zhang, and Julian McAuley. 2023. Personalized Showcases: Generating Multi-Modal Explanations for Recommendations. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (, Taipei, Taiwan,) *(SIGIR '23)*. Association for Computing Machinery, New York, NY, USA, 2251–2255. https://doi.org/10.1145/3539618.3592036

[14] Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, et al. 2019. Multilingual universal sentence encoder for semantic retrieval. *arXiv preprint arXiv:1907.04307* (2019).

[15] Yelp. 2016. Yelp Dataset. https://www.kaggle.com/yelp-dataset/yelp-dataset.